We Claim:

1. A method for synchronizing texture pipelines in a graphics engine, comprising:

loading polygon state variables into an accumulation portion of a plurality of sets of texture pipeline state variable queues; and

enabling a texture processing portion of a number of the sets of state variable queues corresponding to a number of parallel texture operations indicated by the polygon state variables.

2. The method of claim 1, wherein the loading further comprises for each texture pipeline state variable queue:

receiving the polygon state variables in a state variable accumulator; and

copying the received polygon state variables to a state variable latching register.

3. The method of claim 2, wherein the copying is performed prior to processing each polygon.

4. The method of claim 1, further comprising disabling the texture processing portions of the remaining sets of state variable queues.

5. The method of claim 4, further comprising removing power to the pipelines corresponding to the disabled texture processing portions.

6. A method of synchronizing multiple texture pipelines, comprising:

accumulating state variable data in each texture pipeline;

for a predetermined number N of the texture pipelines, advancing the accumulated state variable data to succeeding portions of the texture pipeline, N representing a number of textures to be applied to polygon data.

7. The method of claim 6, further comprising disabling remaining texture pipelines.

8. The method of claim 6, wherein the accumulating comprises:

receiving new state variable data, the new state variable data being defined differentially with respect to old state variable data previously accumulated, and

evicting obsolete elements of the old state variable data in favor of the new state variable data.

9. A control method for a texture processing system having multiple texture pipelines, comprising:

in a state variable queue, accumulating state variable data in a register, the state variable data being received over a plurality of clock cycles as a plurality of data units;

when the texture processing system switches modes, transitioning from a first number of active texture pipelines to a second number of active texture pipelines; and then, in each of a number of texture pipelines corresponding to the second number,

advancing the state variable data from the respective registers to a remainder of the respective state variable queues, and

disabling the remaining texture pipelines and portions of associated state variable queues.

10. The method of claim 9, further comprising enabling power to the second number of active texture pipelines.

11. A texture processing system, comprising:

a plurality of texture pipelines;

a plurality of sets of state variable queues, one set of state variable queues provided for each texture pipeline, the state variable queues each comprising an accumulation register and a latching register coupled to a series of state variable queue processing stages; and

a controller having a first control output adapted to disable at least one of the series of state variable queue processing stages.

12. The texture processing system of claim 11, wherein said controller has a data input adapted to receive a state variable from a programming source.

13. The texture processing system of claim 12, wherein said controller has a data output adapted to transfer a received state variable to the accumulation register of each state variable queue.

14. The texture processing system of claim 12, wherein said controller has a second control output adapted to trigger the transfer of state variables from the accumulation register to the corresponding latching register of each state variable queue.

15. A computer system, comprising:

a processor coupled to a bus;

a system memory in communication with the bus; and

a graphics processor comprising the texture processing system of claim 11.